Tracking as Segmentation of Spatial-Temporal Volumes by Anisotropic Weighted TV

Markus Unger, Thomas Mauthner, Thomas Pock, and Horst Bischof

Graz University of Technology, Austria
{unger, mauthner, pock, bischof}@icg.tugraz.at
http://www.gpu4vision.org

Abstract. Tracking is usually interpreted as finding an object in single consecutive frames. Regularization is done by enforcing temporal smoothness of appearance, shape and motion. We propose a tracker, by interpreting the task of tracking as segmentation of a volume in 3D. Inherently temporal and spatial regularization is unified in a single regularization term. Segmentation is done by a variational approach using anisotropic weighted Total Variation (TV) regularization. The proposed convex energy is solved globally optimal by a fast primal-dual algorithm. Any image feature can be used in the segmentation cue of the proposed Mumford-Shah like data term. As a proof of concept we show experiments using a simple color-based appearance model. As demonstrated in the experiments, our tracking approach is able to handle large variations in shape and size, as well as partial and complete occlusions.

1 Introduction

Although frequently tackled over the last decades, robust visual object tracking is still a vital topic in computer vision. The need for handling variations of the objects appearance, changes in shape and occlusions makes it a challenging task. Additionally, robust tracking algorithms should be able to deal with cluttered and varying background and illumination variations. We formulate the tracking problem as globally optimal segmentation of an object in the spatial-temporal volume. Under the assumption that an object undergoes only small geometric and appearance changes between two consecutive frames, the object is represented as a connected volume containing similar content. Applying the segmentation on a volume instead of single frames, enhances robustness in the case of partial occlusions and similar background. Furthermore, no explicit shape model has to be learned in advance. Instead spatial and temporal consistency is enforced by a single regularization term.

1.1 Related Work

Numerous different approaches have been applied to the visual tracking problem. For a detailed review we refer to [1]. Superior results have been achieved by patch-based [2] or simple kernel-based methods such as [3]. Avidan [3]

considered tracking as a binary classification problem on the pixel level. An ensemble of weak classifier is trained on-line to distinguish between object and current background, while a subsequent mean-shift procedure [4] obtains the exact object localization. Grabner et al. [2] proposed on-line AdaBoost for feature selection, where the object representation is trained on-line with respect to the current background. Although those methods have shown their robust tracking behavior in several applications, they lack an explicit representation of the objects shape, due to their representation by a simple rectangular or elliptical region. Under the assumption of affine object transformation interest point based trackers, like the work of Ozuysal et al. [5], perform excellent with fast runtimes. The drawback of such approaches is the enormous amount of needed pre-calculated training samples, and the limitation that no update is done during tracking. Shape-based [6] or contour based [7] tracking methods deliver additional information about the object state or enhance the tracking performance on cluttered background. While Donoser and Bischof [6] used MSER [8] segmentation results for tracking, Isard and Blake [7] applied the CONDENSATION algorithm on edge information. Therefore feature extraction or segmentation were independent from the tracking framework. In contrast, especially level-set methods support the unified approach of tracking and segmentation in one system [9], [10], [11], [12], [13]. [10] modeled object appearance using color and texture information while a shape prior is given by level sets. [11] incorporated Active Shape Model based on incremental PCA, which allowed the online adoption of the shape models. [9] extended the mean-shift procedure by [4], by applying fixed asymmetric kernels to estimate translation, scale and rotation. For a more detailed review on the use of level set segmentation we refer to [12]. Recently, Bibby et al. [13] proposed an approach, where they used pixel-wise posterior instead of likelihoods in a narrow band level set framework for robust visual tracking. The use of pixel-wise posterior led to sharper extrema of the cost function, while the GPU based narrow band level set implementation achieved real-time performance. All of the above approaches work on single frames. In [14], Mansouri et al. proposed a joint space-time segmentation algorithm based on level sets. The main idea of interpreting tracking as segmentation in a spatial-temporal volume is closely related to the approach presented in this paper. In contrary to our approach level set methods are used, that can easily get stuck in local minima.

A lot of work has been done on image segmentation. For contour-based image segmentation the Geodesic Active Contour (GAC) model [15] has received much attention. In the following we will shortly review some energy minimization based approaches. Graph cuts are currently widely used for computer vision applications. Boykov et al. [16], [17] used a minimum cut algorithm to solve a graph based segmentation energy. Other graph based segmentation approaches were proposed by Grady with the random walker algorithm [18], which was extended in [19]. In [20], a TV based energy was used for segmentation of moving objects. While graph cuts allow simple and fast implementations, it is well-known that the quality of the segmentation depends on the connectivity of the underlying

graph, and can cause systematic metrication errors [21]. Furthermore memory consumption is usually very high. Continuous maximal flows were presented by Appleton et al. [22]. In [23], Zach et al. extended continuous maximal flows to the anisotropic setting.

Variational approaches try to obtain a segmentation based on a continuous energy formulation. Therefore the weighted Total Variation (TV) as used by Bresson et al. [24], [25], Leung and Osher [26] and Unger et al. [27], [28], has become quite popular. Continuous formulations do not suffer from metrication errors, and have become reasonable fast by implementing them on the GPU [28]. Another well known variational segmentation framework is the Mumford-Shah image segmentation model [29]. Bresson et. al. [30] showed how non-local image information can be incorporated into a variational segmentation framework. In [31], Werlberger et al. showed how shape prior information can be incorporated using a Mumford-Shah like data term.

2 Tracking as Segmentation in a Spatial-Temporal Volume

In the following we will provide some details on the concept of interpreting tracking as the segmentation of a 3D volume similar to [14]. A color image I is defined in the 2D image domain Ω as $I : \Omega \rightarrow \mathbb{R}^3$. The 2D frames of a video sequence can be viewed as a volume by interpreting the temporal domain T as the third dimension. Thus the volume is defined as $V : (\Omega \times T) \rightarrow \mathbb{R}^3$. This makes it possible to incorporate spatial and temporal regularization in an unified framework. If we assume a high enough sampling rate, adjoining frames will contain similar content. The 2D objects of a single frame I correspond to cuts of planes with the 3D object defined in the volume V. Inherently this approach extends the forward propagation of information through time by additional backward propagation. Objects that are represented as disjoint regions in a single frame, correspond to a single volume, and are therefore tracked robustly. This concept is illustrated with an artificial example in Figure 1. Our tracking approach is compared to an MSER tracker [6], that cannot handle multiple disjoint regions. The volumetric approach does not suffer from such a shortcoming, as the regions are connected in the volume.

3 Algorithm

3.1 The Segmentation Model

We propose to use the following variational minimization problem for the task of image segmentation:

$$\min_u \left\{ E_p = \int_{\Omega \times T} \left(g_x |\nabla_{\boldsymbol{x}} u| + g_t |\nabla_t u| \right) d\boldsymbol{x} dt + \lambda \int_{\Omega \times T} f u \; d\boldsymbol{x} dt \right\} . \qquad (1)$$

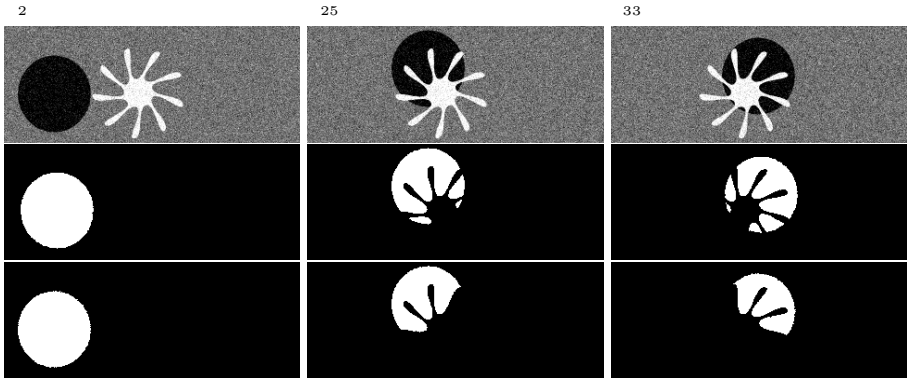

Fig. 1. Tracking of an artificial object. The first row depicts frames of the input video with frame numbers at the top. The second row shows the segmentation result using the volumetric approach. The third row shows the result of an MSER tracker implementation [6].

The first term is a regularization term using anisotropic TV. The segmentation is represented by $u : (\Omega \times T) \rightarrow [0, 1]$. A binary labeling into foreground F ($u = 1$) and background B ($u = 0$) would force $u \in \{0, 1\}$. As this would make the energy non-convex, we can make use of convex relaxation [32]. For the g-weighted TV, Bresson already showed [33] that by letting u vary continuously, the regularization term becomes convex. To obtain a binary segmentation, any levelset of u can be selected using thresholding [23]. The segmentation cue $f : (\Omega \times T) \rightarrow \mathbb{R}$ gives hints whether the pixel belongs to the foreground or the background. The gradient operators in the regularization term are defined as $|\nabla_{\boldsymbol{x}} u| = \sqrt{\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2}$ in the spatial domain Ω, and $|\nabla_t u| = \left|\frac{\partial u}{\partial z}\right|$ in the temporal domain T. Edge information is incorporated by $g_x : (\Omega \times T) \rightarrow \mathbb{R}$ and $g_t : (\Omega \times T) \rightarrow \mathbb{R}$ that subsequently represent edges in the current frame and edges from one to the next frame. The edge potential g_x is computed as $g_x = \exp\left(-a \left|\nabla_{\boldsymbol{x}} V\right|^b\right)$. Likewise one can compute $g_t = \exp\left(-a \left|\nabla_t V\right|^b\right)$. The edge detection function maps strong edges to low values. Consequently discontinuities in u that correspond to the image region, are likely to be located at low values of g_x and g_t during the minimization process. This ensures that the segmentation boundary snaps to strong edges in the image.

The Mumford-Shah [29] like data term was already used in [31] for shape prior segmentation. For the segmentation cue f we distinguish the following cases: If $f = 0$ the data term is eliminated and segmentation is done solely based on edges. If $f > 0$ the segmentation cue gives a background hint. The bigger the value of f, the more likely it will be classified as background. In a similar manner $f < 0$ gives foreground hints. We use color features as described

in Section 4.2 to compute f. Of course any other features or information can be incorporated through the segmentation cue.

The usage of an anisotropic weighted TV norm for regularization has the advantage that discontinuities in the spatial domain V and in the time domain T are separated. This allows a more accurate segmentation of small and fast moving objects. To illustrate this, Figure 2 shows a comparison of the regularization term as used in (1), and the standard weighted TV as used in [33] and [28]. Therefore we simply replaced the regularization term by $\int_{\Omega\times T} g|\nabla u|\ d\boldsymbol{x}dt$ with $g = \exp\left(-a\left|\nabla V\right|^b\right)$. It shows that the anisotropic regularization delivers finer details during fast moving parts of the video.

Fig. 2. Comparison of anisotropic TV and standard TV regularization. The first row shows frames of the original video where the left player is tracked. In the second row the anisotropic regularization term shows a better segmentation of fast moving details than the standard weighted TV regularization in the third row.

3.2 Solving the Minimization Problem

In the following we derive an adaption of the primal-dual algorithm of Zhu et al. [34]. To solve the energy defined in (1), we use duality by introducing the dual variable $\boldsymbol{p} : \Omega \times T \to \mathbb{R}^3$. The dual variable can be separated into a spatial and a temporal component $\boldsymbol{p} = (\boldsymbol{p_x}, p_t)^T$. Thus we get the following constrainted primal-dual formulation of the segmentation model:

$$\min_u \left\{ \sup_{\boldsymbol{p}} \left\{ E_{pd} = -\int_{\Omega\times T} u\nabla\cdot\boldsymbol{p}\ d\boldsymbol{x}dt + \lambda\int_{\Omega\times T} f u\ d\boldsymbol{x}dt \right\} \right\} \tag{2}$$

$$s.t.\quad |\boldsymbol{p_x}(\boldsymbol{x},t)| \le g_x(\boldsymbol{x},t),\quad |p_t(\boldsymbol{x},t)| \le g_t(\boldsymbol{x},t)\ . \tag{3}$$

The dependence on ∇u in the primal energy $E_p(\boldsymbol{x}, t, u, \nabla u)$ is removed in the primal dual energy $E_{pd}(\boldsymbol{x}, t, u, \boldsymbol{p})$, but the problem is now an optimization problem in two variables. This energy can be solved using alternating minimization with respect to u and maximization with respect to $\boldsymbol{p}$.

When updating the primal variable u (primal update) we derive (2) according to u and arrive at the following Euler-Lagrange equation:

$$-\nabla \cdot \boldsymbol{p} + \lambda f = 0 \,. \tag{4}$$

Performing a gradient descent update scheme this leads to

$$u^{n+1} = \Pi_{[0,1]}\big(u^n - \tau_p\left(-\nabla \cdot \boldsymbol{p} + \lambda f\right)\big) \,, \tag{5}$$

with τ_p denoting the timestep. The projection Π towards the binary set $[0, 1]$ can be done with a simple thresholding step.

In a second step we have to update the dual variable $\boldsymbol{p}$ (dual update). Deriving (2) according to $\boldsymbol{p}$ one gets the following Euler-Lagrange equation:

$$\nabla u = \mathbf{0} \tag{6}$$

with the additional constraints on $\boldsymbol{p_x}$ and p_t as defined in (3). This results into a gradient ascent method with a trailed re-projection to restrict the length of $\boldsymbol{p}$:

$$\boldsymbol{p}^{n+1} = \Pi_C\big(\boldsymbol{p}^n + \tau_d \nabla u\big) \tag{7}$$

Here the convex set $C = \left\{\boldsymbol{q} = (\boldsymbol{q_x}, q_t)^T : |\boldsymbol{q_x}| \leq g_x, |q_t| \leq g_t\right\}$ denotes a cylinder centered at the origin with the radius g_x and height g_t. The re-projection onto C can be formulated as

$$\Pi_C(\boldsymbol{q}) = \left(\frac{\boldsymbol{q_x}}{\max\left\{1, \frac{|\boldsymbol{q_x}|}{g_x}\right\}}, \max\left\{-g_t, \min\left\{q_t, g_t\right\}\right\}\right)^T \tag{8}$$

Primal (5) and dual (7) updates are iterated until convergence. As u is a continuous variable, and the energy in (1) is not strictly convex, u may not be a binary image. Any level set of u can be selected as a binary segmentation by applying a threshold $\theta \in [0, 1]$. We left $\theta = 0.5$ throughout this paper. An upper boundary for the timesteps can be stated as $\tau_d \tau_p \leq \frac{1}{6}$. In conjunction with [34], an iterative timesteps schema was chosen as:

$$\tau_d(n) = 0.3 + 0.02n \,, \tag{9}$$

$$\tau_p(n) = \frac{1}{\tau_d(n)}\left(\frac{1}{6} - \frac{5}{15+n}\right) \,, \tag{10}$$

where n is the current iteration.

As a convergence criterion the primal-dual gap is taken into account [34]. The primal energy E_p was already defined in (1). For the dual energy E_d we

have to reformulate the primal-dual energy (2). For a fixed $\boldsymbol{p}$, the minimization problem of u can be determined as:

$$u(\boldsymbol{x},t) = \begin{cases} 1 \text{ for } & -\nabla \cdot \boldsymbol{p}(\boldsymbol{x},t) + \lambda f(\boldsymbol{x},t) < 0 \\ 0 \text{ else} & \end{cases} \tag{11}$$

Thus the dual energy can be written as

$$E_d = \int_{\Omega \times T} \min\left\{-\nabla \cdot \boldsymbol{p} + \lambda f, 0\right\} d\boldsymbol{x}dt \ . \tag{12}$$

As the optimization scheme consists of a minimization and a maximization problem, E_p presents an upper boundary of the true minimizer of the energy, and E_d presents a lower boundary. The primal-dual gap is defined as

$$G\left(u, \boldsymbol{p}\right) = E_p\left(u\right) - E_d\left(\boldsymbol{p}\right) \ . \tag{13}$$

An automatic convergence criterion can be defined based on the normalized primal-dual gap, as

$$\lambda \left| \frac{G\left(u, \boldsymbol{p}\right)}{E_p\left(u\right)} \right| < \zeta \ , \tag{14}$$

with ζ the convergence threshold. It showed throughout the experiments, that $\zeta = 0.06$ is a good choice for the convergence threshold.

4 Implementation

4.1 The Segmentation Framework

Due to limitations in computer hardware such as memory, the size of volumes that can be computed at once is limited. Although modern computing hardware can handle volumes with several thousand frames, the necessity of working on the complete sequence at once restricts tracking to offline data. Multiple similar objects, or disjoint regions belonging to the same object (e.g. by occlusions) make additional information necessary. When attempting a general framework with objects of arbitrary size and shape, this becomes a difficult task.

To tackle these problems, we propose to use an incremental approach. Only n frames are segmented at once. The algorithm is initialized on the first n frames, e.g. by drawing a rectangle around the desired object. See Section 4.2 for details of the feature based segmentation approach. If multiple objects are segmented, the user can select the desired object manually. After convergence of the segmentation algorithm (Section 3.2) foreground and background models are updated. Next, the oldest $m < n$ frames are discarded, and m new frames are added to the volume V. To speed up the tracking process we compute the segmentation only on small areas around the current object. To prevent the algorithm from segmenting similar nearby objects, only regions that overlap with the segmentation mask of the last step are selected. In case of occlusions the volumetric representation of an object might be separated into several disjoint regions. Our

overlap constraint causes the tracker to discard the new region. To handle occlusions in general we therefore use the following strategy: We keep track of the average region size. If the segmentation gets smaller than a certain percentage of this average region size, the object is assumed to be occluded. In case of an occlusion the region we are working on starts to grow slowly, and no updates of the foreground and background model are done. If a region is segmented that is big enough to be considered as the object, tracking is continued on this region. Any slice $k \in [1, n]$ of the volume V can be selected as the tracking result. The number of frames the tracker looks into the future is defined by $n - k$. Thus the smaller k and the bigger n, the more robust disjoint regions are tracked.

Implementation of the tracker was done mainly on the GPU using the CUDA framework [35]. The volume depth was fixed for all experiments to $n = 8$, while slice $k = 4$ was used for the segmentation result.

4.2 Color Tracking

Object appearance is represented in RGB color space using a foreground histogram $H_F : \mathbb{R}^3 \rightarrow [0, 1]$, and a background histogram $H_B : \mathbb{R}^3 \rightarrow [0, 1]$. Following the ideas presented in [13], we are using the pixel-wise posterior instead of modeling the color appearance using the likelihood like e.g. [36]. We define $M = M_F, M_B$ as the model parameter that is either foreground F or background B. From the initialization, we obtain the foreground and background likelihoods $P(H_F|M_F)$ and $P(H_B|M_B)$. Applying Bayesian rule we can estimate the posterior $P(M_F|H_F)$ of a pixel being foreground in the context of the actual background given by $P(H_B|M_B)$ and a region-prior $P(M_j)$ with $j \in F, B$ by:

$$P(M_F|H_F) = \frac{P(H_F|M_F)P(M_F)}{\sum_{j=F,B} P(H_j|M_j)P(M_j)} \tag{15}$$

We keep track of foreground and background models by updating them online using an adaption rate α with likelihoods estimated from the current frame $P_{new}(H_j|M_j)$ as:

$$P(H_j|M_j) = (1 - \alpha)P_{old}(H_j|M_j) + \alpha P_{new}(H_j|M_j) \quad with \quad j \in F, B \tag{16}$$

In contrast to [13] we do not apply marginalization. Instead we simply set the segmentation cue $f(\boldsymbol{x}, t) = 0.5 - P(M_F|H_F(V((\boldsymbol{x}, t))))$.

5 Experimental Results

The videos presented in this Section and the software binaries are available online at http://www.gpu4vision.org.

In Figure 3, a white cat is successfully tracked and segmented. The first row shows the input video with different overlays. The rectangle is indicating the

current working region. The blue color of the rectangle indicates that the tracker is working normally. If the object is believed to be occluded in some slice, the rectangle becomes orange. The current object is indicated by an orange overlay. If parts of the image get segmented, but do not belong to the object, these areas are indicated in red. Note that some regions are segmented that do not belong to the object, but most of the incorrect regions are removed. The second row shows the segmentation cue f where the value -0.5 is mapped to black and indicates foreground, the value 0.5 is mapped to white and indicates background. Frame 409 shows a segment where a cross-fade occurs. The tracker detects the loss of the object, starts growing the search region and begins to search for the object. Frame 418 shows that the object was found correctly. Also note that the algorithm always correctly tracks the object despite large scale changes, as our tracking approach makes no restrictions on the region size.

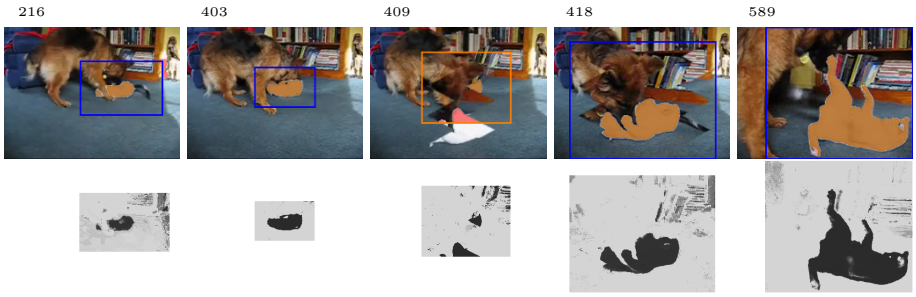

Fig. 3. Tracking example of a cat. The first row depicts the tracked object with the current segmentation and the working region as overlays to the original input image. In the second row the segmentation cue f is depicted in the range $[-0.5,\ 0.5]$.

The second example presented here shows the tracking of a fish in an aquarium. In the top row of Figure 4, the input video is shown, while the bottom row shows the extracted fish. Note that although several partially and complete occlusions occur, the tracker does not loose the object throughout the video. In case of partial occlusions the fish is still correctly segmented, as can be seen in frames 438 and 487. Also note that large shape changes do cause tracking failures, as we make no assumption on shape. In Figure 5, the video is displayed as a volume. The region corresponding to the fish is rendered using iso-surface rendering based on the segmentation mask as obtained by the tracker.

Naturally a color based tracker without any restrictions on shape and scale has its limitations. In Figure 6 a player in a volleyball game is tracked. In the beginning the tracker starts very promising by separating skin tones from the very similar sand. Around frame 337 the skin tones of other players appear in the working region, and are learned as background. As one can see in frame 377 the tracker looses the legs and arms, but still tracks the very characteristic green shirt. In frame 551 the player gets occluded by his team member, with a very

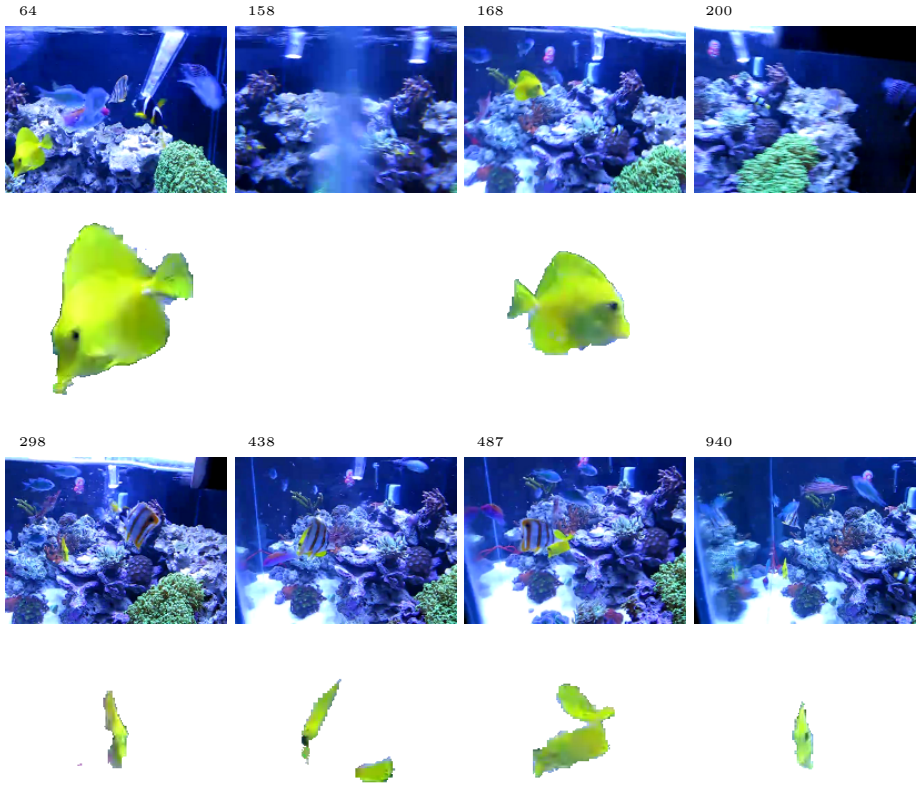

Fig. 4. Tracking sequence of a fish in an aquarium, showing the ability of the tracker to handle large changes in shape, and various kinds of occlusions. The first row depicts the input video and the second row the extracted object.

similar appearance. As no additional high level information is available, both players are tracked.

In Figure 7, another video sequence is shown where the tracker fails. We tried to track the skin of the person. Due to the many occlusions the volume corresponding to skin is separated into several disjoint regions, causing problems for the tracker. Though the tracker can recover several times, the object is permanently lost in frame 276. Other reasons for the failure in this video is the bad discrimination of foreground and background by using solely color.

Experimental results showed that a simple color tracker benefits form interpreting tracking as segmentation in 3D. The tracker successfully handled large variations in scale and shape. The examples show, that the tracker can deal with partial occlusions. Due to the incremental approach also long complete occlusions do not oppose any problem to the tracker. Figure 6 shows an improtant

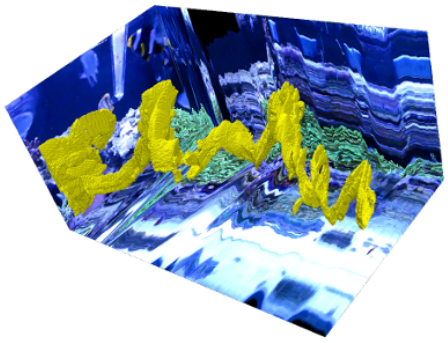

Fig. 5. A schematic 3D rendering of the fish tracking sequence from Figure 4. The tracking result is rendered in yellow.

Fig. 6. Tracking of volleyball sequence, where tracker fails due to highly similar object and colors in the background. The first row shows the input video and the bottom row shows the extracted player.

characteristic of the tracker to adapt foreground and background models to the most characteristic color values. This has the advantage of making the track-

Fig. 7. Video example where tracking and segmentation fail, due to too many occlusions, and bad discrimination of the color histograms.

ing of the object more robust, but also decreases segmentation performance. It also showed that multiple objects with similar appearance cannot be kept apart if occlusions occur. Here clearly high level information could help, e.g. in the volleyball example restrictions on the region size could be made, and shape information would definitely improve results.

6 Conclusion and Future Work

We presented a tracking approach that tracks objects by segmenting them in a spatial-temporal volume. By using the segmentation result a pixel wise classification into foreground and background is achieved. The volumetric tracker presented in this paper, shows promising results for the examples provided in Section 5. An incremental tracking approach was presented and implemented, that works only on a small volume at a time, eliminating memory problems and allowing tracking of videos of arbitrary length. Due to the segmentation in a 3D volume, information is also propagated back through time if the regions are connected in 3D, showing improvements for tracking disjoint regions. As we make no assumptions on shape or scale even large variations cause no problems to the tracker. The tracker is able to handle partial as well as complete occlusions. It was shown that a pure color based foreground and background description is sometimes not sufficient, and leaves room for further improvement.

Future work will focus on more robust modeling of foreground and background regions. Texture features or patches would certainly improve segmenta-

tion and tracking results. Furthermore more complex appearance models with spatial modeling could improve the tracker significantly. Moreover we will focus on a more efficient implementation to achieve near realtime performance.

References

1. Yilmaz, A., Javed, O., Shah, M.: Object tracking: A survey. ACM Comput. Surv. **38**(4) (2006) 13
2. Grabner, H., Grabner, M., Bischof, H.: Real-time tracking via on-line boosting. In: British Machine Vision Conference. (2006) 47–56
3. Avidan, S.: Ensemble tracking. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition. (2005) 494–501
4. Comaniciu, D., V., R., Meer, P.: Real-time tracking of non-rigid objects using mean shift. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition. (2000) 142–149
5. Ozuysal, M., Fua, P., Lepetit, V.: Fast keypoint recognition in ten lines of code. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition. (June 2007) 1–8
6. Donoser, M., Bischof, H.: Efficient maximally stable extremal region (MSER) tracking. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition. (2006) 553–560
7. Isard, M., Blake, A.: Contour tracking by stochastic propagation of conditional density. In: Proc. European Conference on Computer Vision. (1996) 343–356
8. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions. In: British Machine Vision Conference. (2002) 384–393
9. Yilmaz, A.: Object tracking by asymmetric kernel mean shift with automatic scale and orientation selection. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition. (2007) 1–6
10. Yilmaz, A., Li, X., Shah, M.: Contour based object tracking with occlusion handling in video acquired using mobile cameras. IEEE Trans. on Pattern Analysis and Machine Intelligence **26** (2004) 1531–1536
11. Fussenegger, M., Roth, P., Bischof, H., Deriche, R., Pinz, A.: A level set framework using a new incremental, robust active shape model for object segmentation and tracking. Image and Vision Computing in press.
12. Cremers, D., Rousson, M., Deriche, R.: A review of statistical approaches to level set segmentation: Integrating color, texture, motion and shape. International Journal of Computer Vision **72** (2007) 195–215
13. Bibby, C., Reid, I.: Robust real-time visual tracking using pixel-wise posteriors. In: Proc. European Conference on Computer Vision. Volume 2. (2008) 831–844
14. Mansouri, A.R., Mitiche, A., Aron, M.: PDE-based region tracking without motion computation by joint space-time segmentation. In: Proc. International Conference on Image Processing. (Sept. 2003) III–113–16 vol.2
15. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. Intl. J. of Computer Vision **22**(1) (1997) 61–79
16. Boykov, Y., Jolly, M.P.: Interactive organ segmentation using graph cuts. In: Proc. of MICCAI 2000, LNCS 1935, Springer (2000) 276–286
17. Boykov, Y., Kolmogorov, V.: Computing geodesics and minimal surfaces via graph cuts. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition. (2003) 26–33

18. Grady, L.: Random walks for image segmentation. IEEE Trans. on Pattern Analysis and Machine Intelligence **28**(11) (Nov. 2006) 1768–1783
19. Sinop, A.K., Grady, L.: A seeded image segmentation framework unifying graph cuts and random walker which yields a new algorithm. In: Proc. International Conference on Computer Vision. (Oct. 2007)
20. Ranchin, F., Chambolle, A., Dibos, F.: Total variation minimization and graph cuts for moving objects segmentation. In: Scale Space and Variational Methods in Computer Vision. (2008) 743–753
21. Klodt, M., Schoenemann, T., Kolev, K., Schikora, M., Cremers, D.: An experimental comparison of discrete and continuous shape optimization methods. In: Proc. European Conference on Computer Vision. (2008) 332–345
22. Appleton, B., Talbot, H.: Globally minimal surfaces by continuous maximal flows. IEEE Trans. Pattern Analysis and Machine Intelligence **28**(1) (2006) 106–118
23. Zach, C., Niethammer, M., Frahm, J.M.: Continuous maximal flows and Wulff shapes: Application to MRFs. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition. (2009)
24. Bresson, X., Esedoglu, S., Vandergheynst, P., Thiran, J., Osher, S.: Global minimizers of the active contour/snake model. In: Free Boundary Problems (FBP): Theory and Applications. (2005)
25. Bresson, X., Esedoglu, S., Vandergheynst, P., Thiran, J., Osher, S.: Fast global minimization of the active contour/snake model. J. Math. Imaging and Vision **28**(2) (2007) 151–167
26. Leung, S., Osher, S.: Fast global minimization of the active contour model with TV-inpainting and two-phase denoising. In: 3rd IEEE Workshop on Variational, Geometric and Level Set Methods in Computer Vision. (2005) 149–160
27. Unger, M., Pock, T., Bischof, H.: Continuous globally optimal image segmentation with local constraints. In: Computer Vision Winter Workshop 2008, Moravske Toplice, Slovenija (February 2008)
28. Unger, M., Pock, T., Trobin, W., Cremers, D., Bischof, H.: TVSeg - Interactive Total Variation based image Segmentation. In: British Machine Vision Conference 2008, Leeds, UK (September 2008)
29. Mumford, D., Shah, J.: Optimal approximations by piecewise smooth functions and variational problems. Comm. on Pure and Applied Math. **XLII**(5) (1988) 577–685
30. Bresson, X., Chan, T.F.: Non-local unsupervised variational image segmentation models. UCLA CAM Report 08-67 (2008)
31. Werlberger, M., Pock, T., Unger, M., Bischof, H.: A variational model for interactive shape prior segmentation and real-time tracking. In: International Conference on Scale Space and Variational Methods in Computer Vision. (June 2009)
32. Nikolova, M., Esedoglu, S., Chan, T.F.: Algorithms for finding global minimizers of image segmentation and denoising models. SIAM J. on App. Math. **66** (2006)
33. Bresson, X., Esedoglu, S., Vandergheynst, P., Thiran, J.P., Osher, S.: Fast global minimization of the active contour/snake model. In: Journal of Mathematical Imaging and Vision. Volume 28. (2007) 151–167
34. Zhu, M., Wright, S.J., Chan, T.F.: Duality-based algorithms for total variation image restoration. UCLA CAM Report 08-33 (2008)
35. Lindholm, E., Nickolls, J., Oberman, S., Montrym, J.: NVIDIA Tesla: A unified graphics and computing architecture. IEEE Micro **28**(2) (March–April 2008) 39–55
36. Cremers, D.: Dynamical statistical shape priors for level set-based tracking. IEEE Trans. on Pattern Analysis and Machine Intelligence **28**(8) (August 2006) 1262–1273